REMARKS

The Office Action dated November 4, 2004, has been received and carefully noted. The following remarks are submitted as a full and complete response thereto. Claims 1-32 are currently pending in the application and are respectfully submitted for consideration.

In the Office Action, claims 1-32 were rejected under 35 U.S.C. §102(a) as being anticipated by Roth (U.S. Patent No. 6,430,666). Applicants respectfully submit that the presently pending claims recite subject matter that is neither disclosed nor suggested by Roth. The rejection is respectfully traversed for the reasons which follow.

Claim 1, upon which claims 2-4 are dependent, recites a memory management method. The method includes assigning pointers to free memory locations, linking the pointers to one another creating a linked list of free memory locations having a beginning and an end, assigning a free head pointer to a memory location indicating the beginning of free memory locations, assigning a free tail pointer to a memory location indicating the end of free memory locations, assigning an initial data pointer to the memory location assigned to said free head pointer, assigning an end of data pointer to a last data memory location, and assigning the free head pointer to a next memory location linked to said last data memory location assigned to said end of data pointer, wherein the next memory location indicates said beginning of free memory locations.

Claim 5, upon which claims 6-8 are dependent, also recites a memory management method. The method includes the steps of: assigning pointers to free

memory locations, linking the pointers to one another creating a linked list of free memory locations having a beginning and an end, assigning a free head pointer to a memory location indicating the beginning of free memory locations, assigning a free tail pointer to a memory location indicating the end of free memory locations, linking the memory location assigned to the free tail pointer to the memory location assigned to an initial data pointer when memory locations occupied by data are to be indicated as free memory, and assigning the free tail pointer to the last data memory location assigned to the end of data pointer.

Claim 9, upon which claims 10-12 are dependent, recites a memory management system. The system includes a pointer assignor that assigns pointers to free memory locations, a linker that links the pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, a free head pointer assignor that assigns a free head pointer to a memory location indicating the beginning of the linked list of free memory locations, a free tail pointer assignor that assigns a free tail pointer to a memory location indicating the end of the linked list of free memory locations, an initial data pointer assignor that assigns an initial data pointer to the memory location assigned to the free head pointer, and an end of data pointer assignor that assigns an end of data pointer to a last data memory location. The free head pointer assignor assigns said free head pointer to a next memory location linked to said last data memory location assigned to the end of data pointer, and the next memory location indicates the beginning of free memory locations.

Claim 13, upon which claims 14-16 are dependent, also recites a memory management system. The system includes a pointer assignor that assigns pointers to free memory locations, a linker that links said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, a free head pointer assignor that assigns a free head pointer to a memory location indicating said beginning of said linked list of free memory locations, and a free tail pointer assignor that assigns a free tail pointer to a memory location indicating said end of said linked list of free memory locations. The linker links said memory location assigned to said free tail pointer to said memory location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory, and the free tail pointer assignor assigns said free tail pointer to said last data memory location assigned to said end of data pointer.

Claim 17, upon which claims 18-20 are dependent, recites a memory management system. The system includes a pointer assignor means for assigning pointers to free memory locations, a linker means for linking said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, a free head pointer assignor means for assigning a free head pointer to a memory location indicating said beginning of said linked list of free memory locations, a free tail pointer assignor means for assigning a free tail pointer to a memory location indicating said end of said linked list of free memory locations, an initial data pointer assignor means for assigning an initial data pointer to said memory location assigned to said free head pointer, and an

end of data pointer assignor means for assigning an end of data pointer to a last data memory location. The free head pointer assignor means assigns said free head pointer to a next memory location linked to said last data memory location assigned to said end of data pointer, and the next memory location indicates said beginning of free memory locations.

Claim 21, upon which claims 22-24 are dependent, recites a memory management system. The system includes a pointer assignor means for assigning pointers to free memory locations, a linker means for linking said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, a free head pointer assignor means for assigning a free head pointer to a memory location indicating said beginning of said linked list of free memory locations, and a free tail pointer assignor means for assigning a free tail pointer to a memory location indicating said end of said linked list of free memory locations. The linker means links said memory location assigned to said free tail pointer to said memory location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory, and the free tail pointer assignor means assigns said free tail pointer to said last data memory location assigned to said end of data pointer.

Claim 25, upon which claims 26-28 are dependent, recites a memory management device. The device includes a pointer assignor that assigns pointers to free memory locations of a memory, a linker that links said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, a free head pointer

assignor that assigns a free head pointer to a memory location indicating said beginning of said linked list of free memory locations, a free tail pointer assignor that assigns a free tail pointer to a memory location indicating said end of said linked list of free memory locations, an initial data packet pointer assignor that assigns an initial data packet pointer to said memory location assigned to said free head pointer, and an end of data packet pointer assignor that assigns an end of data packet pointer to a last data memory location in said memory. The free head pointer assignor assigns said free head pointer to a next memory location linked to said last data packet memory location assigned to the end of data packet pointer, and the next memory location indicates the beginning of free memory locations.

Claim 29, upon which claims 30-32 are dependent, recites A memory management device. The device includes a pointer assignor that assigns pointers to free memory locations of a memory, a linker that links said pointers to one another thereby creating a linked list of free memory locations having a beginning and an end, a free head pointer assignor that assigns a free head pointer to a memory location indicating said beginning of said linked list of free memory locations, and a free tail pointer assignor that assigns a free tail pointer to a memory location indicating said end of said linked list of free memory locations. The linker links said memory location assigned to the free tail pointer to the memory location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory, and the free tail pointer assignor

assigns the free tail pointer to the last data memory location assigned to the end of data pointer.

As will be discussed below, the prior art reference of Roth fails to disclose or suggest all of the elements of the claims, and therefore fails to provide the features discussed above.

Roth discloses a method of forming a linked list in memory using an address generator. The address generator assigns addresses to each of multiple lists, where each list corresponds to a device which uses the memory. When the address generator has assigned each address location once, a free list is used to track available addresses. The free list is not used until all addresses have been assigned once. A counter is incremented each time an address is assigned; the value of the counter provides the address for a write operation. The free list is not used until the counter has been used to assign all addresses in the memory.

Applicants respectfully submit that Roth discloses a different memory management system than that of the claimed invention. Specifically, Roth fails to disclose or suggest assigning an initial data pointer to a memory location assigned to a free head pointer, as recited in claims 1, 9, 17, and 25. As discussed above, Roth discloses that an address generator is used during initial processing to assign addresses to each of multiple lists. The address generator assigns each of the address locations in the memory only once. Then the free list is used to track available addresses (unused

memory cells). The free list is not used until all addresses have been assigned once (Roth, Column 1, line 67 – Column 2, line 7).

According to the claimed invention, on the other hand, an initial data pointer is assigned to the memory location assigned to the free head pointer. In other words, the present invention assigns the initial data pointer to the head of the free list. No such operation is disclosed by Roth.

For example, referring to figure 3A of the claimed invention, if a first packet of data were to be saved, the initial data pointer would be assigned to memory location 302 where the free head pointer was assigned. In step 408 of figure 4A, the next memory location linked to the memory location assigned to the initial data pointer would be assigned to store more data. If the data requires additional free memory locations, memory locations 304 and 306 would also be assigned to data packet 1. At memory location 306, an end of data pointer would be assigned to this last data location (Specification, Paragraph 0072).

Applicants submit that Roth does not disclose that an initial data pointer is assigned to a memory location where a free head pointer was assigned. As discussed above, an address generator assigns address locations for data upon initial processing and therefore no initial data pointer is used. Moreover, upon subsequent operations Roth also fails to disclose assigning an initial data pointer to a memory location assigned to the free head pointer. Rather, Roth discloses that, when adding from the free list, the current address register is copied with the value stored in the head of the free list and the free list

head is loaded with the contents of the linked list memory at the current address. In this manner, the address which is indicated in the head of the free list becomes the first address of the list upon which an operation is to be performed (Roth, Column 7, lines 23-30). Therefore, even upon subsequent operations, Roth fails to disclose or suggest assigning an initial data pointer to a memory location assigned to the free head pointer.

Furthermore, Applicants note that Roth, unlike the present invention, uses a counter that is incremented each time an address is assigned. The value of the counter is used to assign the address for each write operation. The counter value is output into a current address register, mentioned above (Roth, Column 2, lines 8-15). Prior to adding an entry into a list, the counter associated with the list is checked to determine whether the counter equals zero. When the counter equals zero this indicates that the list is empty and the head of the selected list will be assigned the same value as the current address register of this cycle (Roth, Column 7, lines 41-48).

For at least the reasons discussed above, Applicants respectfully submit that Roth fails to disclose or suggest the elements of claims 1, 9, 17, and 25. Claims 2-4, 10-12, 18-20, and 26-28 are dependent upon claims 1, 9, 17, and 25, respectively. Therefore, Applicants respectfully submit that claims 2-4, 10-12, 18-20, and 26-28 are allowable for at least their dependence upon claims 1, 9, 17, and 25, and for the specific limitations recited therein.

In addition Applicants respectfully submit that Roth fails to disclose or suggest the limitation of linking a memory location assigned to a free tail pointer to a memory

location assigned to an initial data pointer when memory locations occupied by data is to be indicated as free memory, as recited in claims 5, 13, 21, and 29.

Figure 4B of the currently claimed invention illustrates the steps taken when the location of a data packet is indicated as being free memory. First, the memory location assigned to the initial data pointer is linked to the memory location assigned to the free tail pointer. For instance, in figure 3C, packet 1 previously occupied memory locations 302, 306, and 308. In this case, the free tail pointer was assigned to memory location 316. The memory location assigned to the free tail pointer 316 is then linked to the memory location assigned to the initial data pointer, memory location 302. In the next step, the free tail pointer is assigned to the memory location assigned to the end of data pointer (Specification, page 25, paragraphs 0076-0078).

Roth does not disclose or suggest the limitations discussed above. Instead, Roth discloses that "once each address has been assigned to a list, the free list will then begin to track available addresses within memory. Memory locations are added to the free list on remove operations and are removed from the free list on add operations" (Roth, Column 8, line 66 – Column 9, line 3). Roth, however, fails to disclose or suggest linking the memory location assigned to the free tail pointer to the memory location assigned to the initial data pointer when memory locations occupied by data are indicated as free memory, and therefore fails to anticipate the elements of claims 5, 13, 21, and 29.

Therefore, Applicants respectfully submit that Roth fails to disclose or suggest all of the elements of independent claims 5, 13, 21, and 29. Applicants note that claims 6-8,

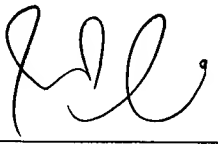
14-16, 22-24, and 30-32 are dependent upon claims 5, 13, 21, and 29, respectively. Therefore, applicants respectfully submit that claims 6-8, 14-16, 22-24, and 30-32 should be allowable for at least their dependence on claims 5, 13, 21, and 29, and the specific limitations recited therein.

Applicants respectfully submit that Roth fails to disclose or suggest critical and important elements of the claimed invention. These distinctions are more than sufficient to render the claimed invention unanticipated and unobvious. It is therefore, respectfully requested that all of claims 1-32 be allowed and this application passed to issue.

If for any reason the Examiner determines that the application is not now in condition for allowance, it is respectfully requested that the Examiner contact, by telephone, the applicants' undersigned attorney at the indicated telephone number to arrange for an interview to expedite the disposition of this application.

In the event this paper is not being timely filed, the applicants respectfully petition for an appropriate extension of time. Any fees for such an extension together with any additional fees may be charged to Counsel's Deposit Account 50-2222.

Respectfully submitted,



Majid S. AlBassam
Registration No. 54,749

Customer No. 32294
SQUIRE, SANDERS & DEMPSEY LLP
14TH Floor
8000 Towers Crescent Drive
Tysons Corner, Virginia 22182-2700
Telephone: 703-720-7800
Fax: 703-720-7802

MSA:jf